





Assessing a Multi-Objective Genetic Algorithm with a Simulated Environment for Energy-Saving of Air Conditioning Systems with User Preferences

Alejandro Humberto García Ruiz ¹, Salvador Ibarra Martínez ¹, José Antonio Castán Rocha ¹, Jesús David Terán Villanueva ^{1,*}, Julio Laria Menchaca ¹, Mayra Guadalupe Treviño Berrones ¹, Mirna Patricia Ponce Flores ^{2,3} and Aurelio Alejandro Santiago Pineda ⁴

- ¹ Facultad de Ingeniería, Universidad Autónoma de Tamaulipas, Tampico 89339, Mexico; a2113330227@alumnos.uat.edu.mx (A.H.G.R.); sibarram@docentes.uat.edu.mx (S.I.M.); jacastan@docentes.uat.edu.mx (J.A.C.R.); jlaria@docentes.uat.edu.mx (J.L.M.); mgtrevino@docentes.uat.edu.mx (M.G.T.B.)
- ² Graduate Program Division, Tecnológico Nacional de México, Madero 89440, Mexico; D94070293@cdmadero.tecnm.mx
- ³ Graduate Program Division, Instituto Tecnológico de Ciudad Madero, Cd., Madero 89440, Mexico
- ⁴ Information Technology Engineering, Polytechnic University of Altamira, Altamira 89602, Mexico; aurelio.santiago@upalt.edu.mx
- Correspondence: jdteran@docentes.uat.edu.mx

Abstract: Electricity is one of the most important resources for the growth and sustainability of the population. This paper assesses the energy consumption and user satisfaction of a simulated air conditioning system controlled with two different optimization algorithms. The algorithms are a genetic algorithm (GA), implemented from the state of the art, and a non-dominated sorting genetic algorithm II (NSGA II) proposed in this paper; these algorithms control an air conditioning system considering user preferences. It is worth noting that we made several modifications to the objective function's definition to make it more robust. The energy-saving optimization is essential to reduce CO2 emissions and economic costs; on the other hand, it is desirable for the user to feel comfortable, yet it will entail a higher energy consumption. Thus, we integrate user preferences with energy-saving on a single weighted function and a Pareto bi-objective problem to increase user satisfaction and decrease electrical energy consumption. To assess the experimentation, we constructed a simulator by training a backpropagation neural network with real data from a laboratory's air conditioning system. According to the results, we conclude that NSGA II provides better results than the state of the art (GA) regarding user preferences and energy-saving.

Keywords: energy optimization; genetic algorithms; multi-objective optimization; artificial neural network simulator

1. Introduction

Due to constant technological growth, it is now possible to integrate electronic circuits and computer systems to develop, manage, and monitor a wide variety of devices in many of the structures that surround us [1–3]. One of the main advantages of including computational techniques in regular devices is the efficient administration of electrical energy. Researchers have focused their efforts on optimizing these regular devices as well as new research focused on the use of artificial intelligence, neural networks, and fuzzy logic and a study on a new class of optimization problems of constrained interval values [4,5]. This paper implements two evolutionary algorithms, a genetic algorithm (GA) and a non-dominated sorting genetic algorithm (NSGA II), to optimize an air conditioning system's energy consumption. We chose to research air conditioning systems because they consume a large amount of energy regardless of the purpose.



Citation: García Ruiz, A.H.; Ibarra Martínez, S.; Castán Rocha, J.A.; Terán Villanueva, J.D.; Laria Menchaca, J.; Treviño Berrones, M.G.; Ponce Flores, M.P.; Santiago Pineda, A.A. Assessing a Multi-Objective Genetic Algorithm with a Simulated Environment for Energy-Saving of Air Conditioning Systems with User Preferences. *Symmetry* 2021, *13*, 344. https://doi.org/10.3390/sym13020344

Academic Editor: Jan Awrejcewic

Received: 27 January 2021 Accepted: 16 February 2021 Published: 20 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). In the literature, it is possible to find some papers focused on ensuring an adequate administration of electrical energy through the so-called home energy management systems which seek to promote renewable energy, lengthen device life, and avoid accidents. In [1], Collota et al. present an example of this system, where a fuzzy controller is used to improve the home energy management scheme; the results showed the system's capability to reduce the maximum load demand for electrical energy. Similarly, Attoue et al. [2] used machine learning techniques, optimization, and data structures to create an energy management system to satisfy the needs of an intelligent building; the results obtained show the proposed mechanisms' effectiveness to improve the performance.

Even though there is a wide variety of papers aiming to save electricity, air conditioners still consume a significant amount of energy. In [6], the authors present a fuzzy controller made in Simulink to lower air conditioning energy while dealing with thermal disturbances. The results demonstrate that the fuzzy controller with triangular membership functions gives the desired performance with an error lower than 1% and saving 25% of energy consumption. Chen, Fu, and Liu [3] proposed a hybrid meta ensemble learning and stacked auto-encoder (SAE), tested with data from the air-conditioning system from a commercial building in Singapore. The proposed meta ensemble learning method showed efficient energy management of the air-conditioning system.

Additionally, Arikiez et al. [7] proposed a heuristic algorithm based on mixed integer linear programming (MILP) to minimize the cost of electrical energy caused by a set of air conditioners. The result reveals that their algorithm can solve a massive problem in a few seconds and gives reasonable suboptimal solutions. Moreover, Ullah and Kim [8] proposed two improved optimization algorithms, a particle swarm optimization and a genetic algorithm, to achieve maximum user comfort in the building environment with minimum energy consumption. Results showed that the proposed optimization algorithms produce better results than the baseline scheme in user comfort and consumed energy.

This paper proposes two metaheuristic algorithms to optimize an air conditioning system while considering the users' preferences, a genetic algorithm (GA) and a non-dominated sorting genetic algorithm (NSGA II). The former intends to replicate the algorithm from [8], and the latter was selected to analyze the performance of a multi-objective optimization algorithm.

The remainder of the paper is as follows: Section 2 addresses the design of the optimization algorithms; Section 3 shows our simulator design; Sections 4 and 5 describe the experimental results and the discussion, respectively. Finally, Section 6 contains the conclusions and future work.

2. Algorithm Design

2.1. Problem Definition

The proposed algorithms aim to tackle two objectives: energy-saving and user satisfaction. These objectives are directly in conflict because the air conditioner system must be turned on for extended periods of time to increase user satisfaction while increasing energy consumption, which conflicts with energy-saving. Such a task seems relatively simple for NSGA II due to its multi-objective nature. Additionally, the genetic algorithm used a weighted objective function to optimize both objectives simultaneously. In this paper, we use the formulation presented in [8] by Ullah and Kim. However, we modify Equations (2)–(4) in two different ways. The first modification adds a cost to maintain the current temperature. The second modification nullifies any energy cost when the laboratory's temperature is lower than the desired temperature. It is important to explain that our climate is warm; therefore, we consider a cooling scheme and turn on the air conditioner system to cool down the room, trying to reach the optimal desired temperature of 22 °C. However, we considered a temperature of 27 °C and below as acceptable yet not no ideal. Furthermore, temperatures below 22 °C are also not desired and would trigger a shutdown of the air conditioner system, nullifying the energy cost and raising the temperature. For this formulation, expression (1) shows the user-accepted temperature range.

$$UserPreferences = [T_{min}, T_{max}]$$
⁽¹⁾

where T_{min} and T_{max} represent the minimum and maximum temperatures accepted by the user. It is worth noting that for our case, T_{min} is the ideal temperature because we are in a tropical environment and a colder climate is desired. Additionally, the current room temperature is T_c , and the temperature recommended by the optimization algorithms is named $T_o|T_{min} \leq T_o \leq T_{max}$.

With T_o , it is possible to calculate the optimal energy (E_o) needed to change or maintain T_c if it were the same as T_o ; see Equation (2). Additionally, if the current temperature is lower than the optimized temperature ($T_c < T_o$), the air conditioner system should be turned off; hence the zero in that case. To calculate this parameter, we need to know the power required per unit change in temperature (P_T).

$$E_{o} = \begin{cases} P_{T} + P_{T} \cdot (T_{c} - T_{o}) & if \ T_{c} \ge T_{o} \\ 0 & if \ T_{c} < T_{o} \end{cases}$$
(2)

Similarly, it is possible to calculate the minimum and maximum hypothetical energy consumptions (E_{min}) and (E_{max}), as shown in expressions (3) and (4), which are used later to calculate the energy-saving gain (G_{es}). For these equations, there is a similar condition as Equation (2); where if $T_c < T_{max}$ or $T_c < T_{min}$, it would mean that the current temperature is lower than the maximum or minimum temperature and the air conditioner system will not need to be turned on to reach those temperatures, hence requiring 0 energy.

$$E_{min} = \begin{cases} P_T + P_T \cdot (T_c - T_{max}) & \text{if } T_c \ge T_{max} \\ 0 & \text{if } T_c < T_{max} \end{cases}$$
(3)

$$E_{max} = \begin{cases} P_T + P_T \cdot (T_c - T_{min}) & \text{if } T_c \ge T_{min} \\ 0 & \text{if } T_c < T_{min} \end{cases}$$
(4)

Equations (5) and (6) show the user satisfaction gain (G_{us}) and the energy-saving gain (G_{es}) where both gains range from 0 to 1, with 0 representing no gain and 1 represents the most gain. Furthermore, both objectives are in direct conflict because increasing user satisfaction entails an increased use of the air conditioner resulting in higher energy cost.

$$G_{us} = \left(\frac{T_{max} - T_o}{T_{max} - T_{min}}\right)^2 \in [0, 1]$$
(5)

$$G_{es} = 1 - \left(\frac{E_o - E_{min}}{E_{max} - E_{min}}\right)^2 \in [0, 1]$$
(6)

Additionally, the user relevance factor for each objective is determined by α_{us} for user satisfaction and α_{es} for energy-saving where $\alpha_{us} + \alpha_{es} = 1$. Therefore, the NSGA II would maximize both objectives considering their relevance factor; see Equations (7) and (8).

$$Maximize(\alpha_{us}G_{uc}) \in [0,1] \tag{7}$$

$$Maximize(\alpha_{es}G_{es}) \in [0,1]$$
(8)

On the other hand, GA uses the weighted function in Equation (9).

$$Maximize(\alpha_{us}G_{uc} + \alpha_{es}G_{es}) \in [0, 1]$$
(9)

Figure 1 shows an example of the calculation of user satisfaction gain (G_{us}) and the energy-saving gain (G_{es}) for a given T_o and T_c .

1:	$T_o = 24.9039$	$T_c = 30.2000$
	$T_{min} = 22$	$T_{max} = 27$
	$P_T = 5$	
2:	$E_o = 5 + 5 \cdot (30.2000 - 24.9039) \\= 31.4805$	
3:	$E_{min} = 5 + 5 \cdot (30.2000 - 27) = 21.00$	$E_{max} = 5 + 5 \cdot (30.2000 - 22) = 46.00$
4:	$G_{us} = \left(\frac{27 - 24.9039}{27 - 22}\right)^2 = 0.1757$	$G_{es} = 1 - \left(\frac{31.4805 - 21.00}{46.00 - 21.00}\right)^2 = 0.8242$

Figure 1. Example of user satisfaction gain (G_{us}) and the energy-saving gain (G_{es}) calculation.

2.2. GA and NSGA II

As we said before, we modify the formal definition of [8], trying to make it more robust by considering new elements; however, we will not be able to compare directly to their results because of the modifications in the definition. Thus, in this paper, we implement their genetic algorithm as explained in [8], and a multi-objective NSGA II as our proposal (both algorithms are available at https://github.com/ahgarciar/Air-Conditioner-System-Controllers (accessed on 19 February 2021)).

We use the same population, selection, crossover, and mutation operators for both algorithms with little modifications for the NSGA II, as explained below.

Population. We use a vector of one hundred solutions for both algorithms, where each solution $(T_{o_i} \in \mathbb{R})$ is a possible temperature between 22 and 27 °C.

Selection. The algorithms use a binary tournament between random individuals of the population to select the parents for the crossover operator. The GA uses a binary tournament, comparing the solutions' objective values, and the winners are selected for crossover. On the other hand, the NSGA II algorithm uses the rank of the Pareto front of each solution to identify the winner, which will be the solution with the lowest level on the Pareto front, meaning that it is less dominated than the other solution. However, suppose both solutions are in the same level of the Pareto front. In that case, we use the crowded comparison operator ($>_n$), which identifies the solution with the largest crowding distance, which will be explained further in this section.

Crossover. As stated before, our GA and NSGA II have a static population of one hundred solutions; then, another one hundred solutions are produced as offspring, using the Simulated Binary Crossover operator [9]. This operator requires two parents (*p*1 and *p*2) to obtain two new individuals (T_{o_i} and $T_{o_{i+1}}$).

Here, we explain the crossover operator for our specific problem. Thus, we will change some of the original notation of this crossover to match our variables.

First, we identify the two parents' maximum and minimum temperature values; see Equations (10) and (11).

$$max = MAX(p1, p2) \tag{10}$$

$$min = MIN(p1, p2) \tag{11}$$

Then, we calculate Equations (12) and (13) to produce $\beta 1$ and $\alpha 1$, which also use a random real number ($u \in [0, 1]$) and a constant $\eta = 20$ in Equation (14) to produce $\beta 1q$. Finally, we produce T_{o_i} as shown in Equation (15).

$$\beta 1 = 1 + \frac{2(\min - T_{\min})}{\max - \min} \tag{12}$$

$$\alpha 1 = 2 - (\beta 1)^{-(\eta + 1)} \tag{13}$$

$$\beta 1q = \begin{cases} (u \cdot \alpha 1)^{\frac{1}{\eta+1}} & \text{if } u \leq \left(\frac{1}{a1}\right) \\ \left(\frac{1}{2-u \cdot \alpha 1}\right)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases}$$
(14)

$$T_{o_i} = 0.5(max + min - \beta 1q * (max - min))$$
⁽¹⁵⁾

Then, we carry out a similar process to produce $T_{o_{i+1}}$. However, the difference is that $\beta 2$ is calculated using T_{max} instead of T_{min} as $\beta 1$, see Equations (16)–(19).

$$\beta 2 = 1 + \frac{2(T_{max} - max)}{max - min} \tag{16}$$

$$\alpha 2 = 2 - (\beta 2)^{-(\eta + 1)} \tag{17}$$

$$\beta 2q = \begin{cases} (u \cdot \alpha 2)^{\frac{1}{\eta+1}} & \text{if } u \leq \left(\frac{1}{\alpha 2}\right) \\ \left(\frac{1}{2-u \cdot \alpha 2}\right)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases}$$
(18)

$$T_{o_{i+1}} = 0.5(max + min - \beta 2q * (max - min))$$
(19)

Figure 2 shows an example of the crossover operator.

1:	p1 = 25.1457	p2 = 25.2698
2:	min = 25.1457	max = 25.2698
3:	$\beta 1 = 1 + \frac{2(25.1457 - 22)}{25.2698 - 25.1457} = 51.6962$	$\beta 2 = 1 + \frac{2(27 - 25.2698)}{25.2698 - 25.1457} = 28.8839$
4:	$\alpha 1 = 2 - (51.6962)^{-(20+1)} = 2.00$	$\alpha 2 = 2 - (28.8839)^{-(20+1)} = 2.00$
5:	Considering $u = 0.9504$	
6:	$\beta 1q = \left(\frac{1}{2 - 0.9504 * 2.00}\right)^{\frac{1}{20+1}} = 1.1164$	$\beta 1q = \left(\frac{1}{2 - 0.9504 * 2.00}\right)^{\frac{1}{20+1}} = 1.1164$
7:	$T_{o_i} = 0.5(25.2698 + 25.1457 - 1.1164 * (25.2698 - 25.1457)) = 25.1384$	$T_{o_{i+1}} = 0.5(25.2698 + 25.1457 + 1.1164 * (25.2698 - 25.1457)) = 25.2770$

Figure 2. Example of crossover process.

Mutation. As a mutation operator, we used the polynomial mutation operator [10], which is applied to fifty percent of the solutions generated in the crossover operator.

Here, we modify the temperature produced by the crossover operation depending on the δ value, which ranges from -1 to 1, and also uses a random real number ($u \in [0, 1]$) and a constant $\eta = 20$; see Equations (20)–(23).

$$T_{o_i} = T_{o_i} + \delta_q (T_{max} - T_{min}) \tag{20}$$

$$\delta_1 = (T_{o_i} - T_{min}) / (T_{max} - T_{min})$$
(21)

$$\delta_2 = (T_{max} - T_{o_i}) / (T_{max} - T_{min}) \tag{22}$$

$$\delta_{q} = \begin{cases} \left[2u + (1.0 - 2.0 * u) * (1.0 - \delta_{1})^{\eta + 1} \right]^{\frac{1}{(\eta + 1)}} - 1 & \text{if } u \le 0.5\\ 1 - \left[2(1 - u) + 2.0 * (u - 0.5) * (1.0 - \delta_{2})^{\eta + 1} \right]^{\frac{1}{(\eta + 1)}} & \text{otherwise} \end{cases}$$
(23)

Figure 3 shows an example of the mutation operator.

1: $T_{o_i} = 25.7195$ 2: $\delta_1 = \frac{25.7195 - 22}{(27 - 22)} = 0.7439$ $\delta_2 = \frac{27 - 25.7195}{(27 - 22)} = 0.2561$ 3: Considering u = 0.31844: $\delta_q = [2 * 0.3184 + (1.0 - 2.0 * 0.3184) * (1.0 - 0.7439)^{20+1}]^{\frac{1}{(20+1)}} - 1 = -0.0212$

Figure 3. Example of the mutation process.

For this experimentation, we tested several values for constant η , resulting in a value of twenty.

The NSGA II requires two additional procedures:

Fast Non-Dominated Sort. This procedure allocates the solutions of the pool of solutions (*P*) into Pareto fronts of non-dominated individuals; this allows elitism in selecting the parents for the offspring of NSGA II.

Crowding Distance. A particular solution's crowding distance value is the average distance of its two neighboring solutions [11]. Figure 4 shows the calculation of the crowding distance of the solution *i*, which is an estimate of the size of the largest cuboid enclosing *i* without including any other solution. The algorithm used in this implementation is a normalized version of the original algorithm with the minimum (MIN_m) and maximum (MAX_m) values per objective (*m*) [12]; this algorithm is calculated for each generation for the current population. The algorithm requires a set of non-dominated individuals (l) to calculate their distance.



Figure 4. Crowding distance calculation.

3. Simulator

To assess the optimization algorithms, we constructed a neural-network-based simulator with temperature data from a specific room within our laboratory. The data acquired was the internal and external temperature (in Celsius) and the state of the air conditioner system. We recorded this data every five minutes using two temperature sensors (DHT22) with an Arduino Mega controller in the second week of September from Monday to Wednesday. Figure 5 shows a diagram from the data acquisition to the evaluation of the neural-network-based simulator. Regarding the data acquisition, we force fluctuations in the internal temperature (from the external temperature to 24 °C) during different moments of the day, turning on and off the air conditioning system.



Figure 5. Diagram from data acquisition to the evaluation of the simulator.

Once the data were acquired, we constructed a dataset with the external temperature, internal temperature, air conditioning state, and the next internal temperature taken from the next row. With this dataset, we trained a backpropagation neural network implemented in C# and is available at https://github.com/ahgarciar/Air-Conditioner-System-Controllers (accessed on 19 February 2021). We configured the neural network with three input neurons, ten neurons in a hidden layer, and one output neuron. Additionally, it was tested with the mean average percentage error (MAPE) [10], producing a 1.07% error; this result shows that our simulator has a relatively good performance.

4. Experimental Results

This experimentation aims to assess energy-saving and user satisfaction for five whole days. To do this, we executed both algorithms independently and used the neural network to simulate the temperature changes, given each algorithm's recommendation.

Figure 6 shows the external temperature acquired for five days and the simulated internal temperature obtained by the neuronal-network-based simulator. The simulator requires three parameters to forecast the new internal temperature: current external temperature, current internal temperature, and the state of the air conditioner system; the latest is obtained according to the optimizer recommendation, i.e., if the optimizer recommendation is lower or equal to the internal temperature, the air conditioning system is turned on; otherwise, it is turned off.

To avoid increasing computational costs, we execute the optimization algorithms once every half hour. These experimentations consider three configurations (see Table 1): prioritizing energy-savings (A), a balance between energy-saving and user satisfaction (B), and prioritizing user satisfaction (C). These three configurations constitute the three primary user profiles within our laboratory and, roughly, in any conventional home.

Table 1. Configurations of the user's priorities.

Configuration	α_{us}	α _{es}
А	0.2	0.8
В	0.5	0.5
С	0.8	0.2



Figure 6. Internal temperature simulation based on the optimizer recommendation.

We configured the algorithms with the values from Table 1; then, the algorithms produce their recommendations for five simulated days; where we use as a recommendation the best solution produced by GA, and a single solution from the first Pareto front for the NSGA II. To keep a fair comparison, we used the nadir point of the Pareto front, which is the solution with the lowest Euclidian distance to a hypothetical optimal point in the space. This hypothetical optimal point is (1, 1) because both objectives intend to maximize their gain, and 1 is the largest possible value for both objectives; see Figure 7.



Figure 7. Nadir point calculation.

Figures 8–10 show the temperature recommended by GA and NSGA II for each moment of the day analyzed (every half an hour) for each of the configurations.



Figure 8. Temperature recommended by genetic algorithms using configuration A.



Figure 9. Temperature recommended by genetic algorithms using configuration B.



Figure 10. Temperature recommended by genetic algorithms using configuration C.

Here, we can see that for configuration A, both algorithms recommend higher temperatures than the other configurations. However, GA does recommend lower values than NSGA II. For configuration B, GA's recommendations have larger fluctuations than NSGA II, which mainly maintains its recommendations from 27 to 23.5 °C. Finally, for configuration C, GA recommends the lowest temperature, about 22 °C, while NSGA II increases its fluctuation.

Based on each algorithm's recommended temperature, the simulator turns the air conditioning on or off for half an hour to adjust the room temperature.

Figures 11–13 show the room temperature simulation results by applying the algorithms' recommendation for each configuration.

30



Figure 11. Simulation of temperature according to the recommendation of the genetic algorithms for configuration A.



Figure 12. Simulation of temperature according to the recommendation of the genetic algorithms for configuration B.



Figure 13. Simulation of temperature according to the recommendation of the genetic algorithms for configuration C.

In Figure 11, configuration A, both algorithms have similar behavior. However, NSGA II produces lower values than GA. Figure 12, configuration B, shows that NSGA II has lower temperature values than GA and both algorithms have considerably different behavior. Finally, Figure 13 shows an almost identical behavior for both algorithms with a slight tendency for the NSGA II to produce lower temperatures than GA.

The satisfaction level increases when the room temperature is closer to the minimum user desired temperature T_{min} . As stated before, we live in a warm environment; thus, we

require a cooling scheme for the air conditioning system. Therefore, the algorithm that keeps closer to T_{min} provides greater satisfaction for the user.

Therefore, to measure user satisfaction, we created a satisfaction ratio, defined as the sum of the absolute differences between the current simulated temperature and T_{min} for each moment of the day; this is because values lower than T_{min} are unwanted.

A lower satisfaction ratio implies a higher level of satisfaction. Table 2 presents the satisfaction ratios calculated for each algorithm for all the configurations.

Table 2. Satisfaction ratio for each algorithm.

Configuration	NSGA II	GA
А	698.10343	993.36546
В	316.98898	1058.35429
С	313.84696	313.88824

As we can see, the NSGA II algorithm achieves better performance over the GA in maintaining the room temperature according to the user preferences. For the configuration where the algorithm prioritizes energy-saving (A), the NSGA II outperformed the GA's satisfaction rate by about 295 units. Furthermore, for the second configuration (B), which prioritizes a balance between energy-saving and user satisfaction, the NSGA II outperformed the GA's satisfaction rate by about 741 units. Finally, for the last configuration (C), which prioritizes user satisfaction, both algorithms produced almost the same satisfaction rate.

To measure the energy consumption, we evaluate the maximum and minimum energy consumption by simulating two extreme scenarios for the whole test. The first with the air conditioning system turned off and the second with the air conditioning system turned on (see Figure 14). According to this simulation and during its execution, there was an electrical energy use of 0 and 105 kWh, respectively.



Figure 14. Temperature simulation considering the air conditioning always off and always on.

Table 3 shows calculations of the number of kWh used by all the configurations from Table 1 for both algorithms. Additionally, we estimated the cost of \$2.997 Mexican pesos for each kWh consumed. We took this electric cost from the Mexican Federal Electricity Commission (CFE). Therefore, for the previous extreme scenario, where the air conditioning system is turned on for the whole test, the cost produced would be \$314.69.

Table 3 shows that NSGA II outperformed GA in the total cost of the electrical energy consumption in two of the three configurations. However, in the non-winning configuration, GA improves by a smaller margin to NSGA II than in the other cases.

Configuration	NSGA II		GA	
Configuration -	kWh	Cost	kWh	Cost
Α	83.74032	\$250.97	86.01629	\$257.79
В	94.29362	\$282.60	93.56877	\$280.43
С	89.98913	\$269.70	91.99613	\$275.71

 Table 3. Cost of maintaining the configurations.

5. Discussion

As we can see, NSGA II achieved a better performance than GA regarding user preferences while also outperforming GA in two of three configurations regarding the energy cost.

Although both algorithms adapt to the user preferences and the energy-saving, the NSGA II algorithm provides greater satisfaction. The neural-network-based simulator showed that the room temperature values of the NSGA II were closer to the user ideal temperature for two of three configurations. In this sense, we can conclude that the NSGA II algorithm produced better results than GA to control the air conditioning systems.

It is worth noting that the experimentation results were on a period of five days with a single room in our laboratory. However, by extending the experimentation to a longer time span and more laboratories/rooms, we can easily reach a cost of about \$3700 for just two rooms with the air conditioning turned on for one month. On the other hand, we can decrease this cost to about \$3000 with the NSGA II configured for energy-saving for the same amount of time. This comparison shows that we can reach an energy-saving of about nineteen percent by using the NSGA II optimizer, highlighting the impact of using an intelligent system on electrical energy consumption.

6. Conclusions

In this paper, we tackle the problem of controlling an air conditioning system using evolutive algorithms to increase energy-saving while considering user satisfaction.

Therefore, we assessed the performance of two genetic algorithms to control an air conditioning system. The first algorithm was an implementation of a genetic algorithm (GA) proposed in [8], while the second was our proposed NSGA II algorithm, which is a multi-objective optimization algorithm. We considered the NSGA II an appropriate alternative because the air conditioning system intends to increase energy-saving while considering the user preference. Thus, both objectives are directly opposed, which is a key identifier of a multi-objective problem, making NSGA II an excellent alternative for solving.

Additionally, we proposed a modification to the definition of the problem, enhancing it to consider new elements, including the cost of maintaining the same temperature and the cancellation of the cost of returning to a warmer temperature, which occurs if the room is colder than the desired temperature.

Furthermore, we designed and implemented a neural-network-based simulator to measure both algorithms' performance in equal circumstances. This simulator showed a relatively low mean average percentage error of 1.07.

Regarding the comparison between GA and NSGA II, the latter showed better energysavings and user satisfaction than GA. Therefore, our assumption that a multi-objective algorithm would produce better results for controlling an air conditioning system with two objectives proved right.

Finally, we are currently working on expanding the experimentation by adding other services. As future work, we encourage researchers to develop an intelligent system prototype using NSGA II as a decision-making engine, integrating different configurations of user preferences and profiles and further studying the air conditioning system's energy cost.

Author Contributions: Conceptualization, Methodology, Investigation, A.H.G.R., S.I.M., and J.D.T.V.; Software, A.H.G.R., J.D.T.V. and A.A.S.P.; Validation, J.A.C.R. and J.D.T.V.; Formal Analysis, Resources, J.A.C.R., and J.L.M.; Writing—Original Draft Preparation, A.H.G.R.; Writing—Review & Editing, S.I.M., J.D.T.V., and A.A.S.P.; Data Curation, Visualization, Supervision, M.G.T.B., M.P.P.F.; Project Administration, Funding Acquisition, S.I.M. and J.A.C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by PNPC of the National Council for Science and Technology (CONACyT) of México, and the Universidad Autónoma de Tamaulipas.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available at https://github.com/ ahgarciar/Air-Conditioner-System-Controllers (accessed on 19 February 2021).

Acknowledgments: A.H.G.R. would like to thank the CONACyT for the scholarship awarded. The APC was funded by the Universidad Autónoma de Tamaulipas.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Collotta, M.; Pau, G. A Solution Based on Bluetooth Low Energy for Smart Home Energy Management. *Energies* 2015, *8*, 11916–11938. [CrossRef]
- 2. Attoue, N.; Shahrour, I.; Younes, R. Smart Building: Use of the Artificial Neural Network Approach for Indoor Temperature Forecasting. *Energies* **2018**, *11*, 395. [CrossRef]
- 3. Chen, Y.; Fu, G.; Liu, X. Air-Conditioning Load Forecasting for Prosumer Based on Meta Ensemble Learning. *IEEE Access* 2020, *8*, 123673–123682. [CrossRef]
- 4. Treanță, S. On a Class of Constrained Interval-Valued Optimization Problems Governed by Mechanical Work Cost Functionals. J. Optim. Theory Appl. 2021, 1–12. [CrossRef]
- 5. Treanță, S. Efficiency in uncertain variational control problems. Neural Comput. Appl. 2020, 1–14. [CrossRef]
- Shah, Z.A.; Sindi, H.F.; Ul-Haq, A.; Ali, M.A. Fuzzy Logic-Based Direct Load Control Scheme for Air Conditioning Load to Reduce Energy Consumption. *IEEE Access* 2020, *8*, 117413–117427. [CrossRef]
- Arikiez, M.; Grasso, F.; Kowalski, D.; Zito, M. Heuristic algorithm for minimizing the electricity cost of air conditioners on a smart grid. In Proceedings of the 2016 IEEE International Energy Conference (ENERGYCON), Leuven, Belgium, 4–8 April 2016; pp. 1–6.
- Ullah, I.; Kim, D. An Improved Optimization Function for Maximizing User Comfort with Minimum Energy Consumption in Smart Homes. *Energies* 2017, 10, 1818. [CrossRef]
- 9. Deb, K.; Kumar, A. Real-coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems. *Complex Syst.* **1995**, *9*, 431–454.
- 10. Deb, K.; Goyal, M. A combined genetic adaptive search (GeneAS) for engineering design. Comput. Sci. Inform. 1996, 26, 30-45.
- Raquel, C.R.; Naval, P.C. An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, Washington, DC, USA, 25–29 June 2005; pp. 257–264.
- 12. Pedersen, G.K.M.; Goldberg, D.E. Dynamic Uniform Scaling for Multiobjective Genetic Algorithms. *Comput. Vis.* **2004**, *3103*, 11–23.